AD-A234 555

(2)

AD

AD-E402 185

Technical Report ARFSD-TR-91007

APPLICATION OF SOFTWARE TOOLS IN EMBEDDED COMPUTERS WITHIN MILITARY SYSTEMS TO ENHANCE SOFTWARE RELIABILITY, QUALITY ASSURANCE, AND MAINTENANCE

William R. Turoczy

DTIC
ELECTE
MAY 01 1991
S B D

April 1991

US ARMY
ARMAMENT MUNITIONS
& CHEMICAL COMMAND
ARMAMENT RDE CENTER

U.S. ARMY ARMAMENT RESEARCH, DEVELOPMENT AND ENGINEERING CENTER

Fire Support Armaments Center
Picatinny Arsenal, New Jersey

Approved for public release; distribution is unlimited.

DTIC FILE COPY

91 4 30 063

The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

The citation in this report of the names of commercial firms of commercially available products or systems does not constitute official endorsement by or approval of the U.S. Government.

Destroy this report when no longer needed by any method that will prevent disclosure of contents or reconstruction of the document. Do not return to the originator.

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information Send comments regarding this burden estimate or any other aspect of th is collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project(0704-0188), Washington, DC 20503

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1991	

4. TITLE AND SUBTITLE

APPLICATION OF SOFTWARE TOOLS IN EMBEDDED COMPUTERS WITHIN MILITARY SYSTEMS TO ENHANCE SOFTWARE RELIABILITY, QUALITY ASSURANCE, AND MAINTENANCE

5. FUNDING NUMBERS

6. AUTHOR(S)

William R. Turoczy

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

ARDEC, FSAC
Artillery Armaments Div
ATTN: SMCAR-FSA-P
Picatinny Arsenal, NJ 07806-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

Technical Report ARFSD-TR-91007

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

ARDEC, IMD
STINFO Br
ATTN: SMCAR-IMI-I
Picatinny Arsenal, NJ 07806-5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report presents the application of the cyclomatic complexity requirement within a military system to improve reliability, quality assurance, and maintenance for the embedded computers within the 155 mm self-propelled Howitzer Improvement Program (HIP). The cyclomatic complexity requirement was used in HIP based on research and documentation described in: (1) NBS 500-99, Cyclomatic Complexity Metric and (2) 1985 Reliability and Maintainability Symposium, The Application of Software Test Tools.

Results from the Joint U.S. Army - Israeli Defense Force HIP are impressive and the report highlights significant achievements within the software engineering discipline and applications within embedded computers within military systems.

14. SUBJECT TERMS

Cyclomatic complexity measure Test driver Automatic testing Control graphs
Embedded computer system Software reliability

15. NUMBER OF PAGES

17

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-1 298-102

CONTENTS

DTIC COPY INSPECTED 6

Accession For
NTIS GRA&I ☒
DTIC TAB ☐
Unannounced ☐
Justification

By
Distribution/
Availability Codes

Dist	Avail and/or Special
A-1	

INTRODUCTION

Reliable software for complex computer systems within the military and commercial systems continue to be an area for investigation and improvement (refs 1, through 7). High costs for embedded computers and poor performance, as compared to ambiguous software requirements, offer an opportunity for application of software tools to enhance software reliability, quality, and maintenance.

Research and development work has been completed by McCabe in a complexity measure for the control structure of a computer program by adapting graph theory (refs 4 and 7). The cyclomatic complexity of the graph is related to the number of independent loops in the graph. McCabe relates the cyclomatic complexity to the minimum number of test paths which are needed to pass through all the branches in the graph of the program.

The process, when implemented, can generate the minimum set of tests for exercising the program and assuring software reliability (refs 4, 5, and 7).

More specifically, the design is impacted by forcing a structured approach to the logic, thereby limiting complexity and yielding more robust, reliable software by building in quality. In addition, this process can generate the minimum set of tests necessary to effectively test every path through the program's logic, improving reliability by achieving greater test coverage.

Based on the above research, the 155 mm self-propelled Howitzer Improvement Program (HIP) generated a software complexity requirement for the project. The Joint U.S. Army-Israeli Defense Force HIP contract stated that the software cyclomatic complexity, as defined in the National Bureau of Standards Special Publication 500-99, shall not exceed 9 (ref 7). In addition, the U.S. Army Armament, Munitions and Chemical Command, Product Assurance Directorate at Picatinny Arsenal, developed an automated Complexity Analysis Tool (CAT) to be used in the HIP project (ref 3). The CAT is a quality assurance tool which provides a quantitative measure of software quality, structure, robustness, testability, and maintainability.

BACKGROUND

The 155 mm self-propelled HIP concept is the result of efforts of government and private contractors to define a multiphased, major cannon field artillery weapon system in response to the Heavy Brigade/Division Field Artillery Fire Support Weapon System Mission Element Need Statement (MENS) dated 12 December 1980 (ref 8). The HIP concepts specifically address four areas of system deficiencies: responsiveness, survivability, terminal effects, and RAM (reliability, availability, and maintainability).

The HIP consists of two configurations where one is the U.S. M109 howitzer and the other is the Israeli M109. The U.S. configuration, which is in limited production, is the M109A6 while the Israeli system is the M109A1C.

The HIP software products are being developed by several contractors both in the U.S. and in Israel. The major software products are:

1. Automatic Fire Control System (AFCS)

2. Prognostic and Diagnostic Interface Unit (PDIU)

3. Simplified Test Equipment Expandable (STE-X)

4. Israeli Automatic Fire Control System (AFCS/IS)

5. Institutional Fire Control System Trainer (IFCST)

6. Institutional Maintenance Trainer (IMT)

The HIP self-propelled howitzer is equipped with an AFCS which includes and is supported by computer resources. The AFCS eliminates the need for section personnel to exit the vehicle during emplacement and displacement of the weapon due to a complete on-board gun-laying/reference system.

The need for conventional survey support is eliminated through the incorporation of a computer support Modular Azimuth Position System (MAPS) which includes a Dynamic Reference Unit (DRU). The M109A1C configuration uses the Gun Orientation Navigation System (GONS) developed in Israel as the DRU alternative.

The Display and Control Unit (DCU) is the man-machine interface for the control of the AFCS. This DCU enables the Chief of Section (CS) to make inputs and/or override the AFCS.

Ballistic computations (technical fire direction) are accomplished on-board, using inputs from the Battery Computer System (BCS) to the AFCS. The AFCS computer controls and displays firing data on the DCU. The AFCS controls the electrical/mechanical servo system to drive the cannon to the computer azimuth and elevation.

The M109A6 employs two Single Channel Ground and Airborne Radio Systems (SINCGARS), frequency hopping, VHF radios to execute digital and voice communications. The digital communication is controlled by the AFCS and is a second, vital link in

the process of enhancing survivability and responsiveness. The M109A6 can communicate directly with its Fire Direction Center (FDC) computer or with a variety of target acquisition systems, either directly or by relay through the FDC, to process the messages required to perform its functions. The highly automated communications processing requires no direct intervention by the howitzer crewman. The main functional areas addressed by the digital communications system are fire missions, logistics, and movement.

The PDIU and STE-X components are used primarily as built-in-test (BIT), Built-in-Test-Equipment (BITE), and maintenance support programs to diagnose faults and failures.

In addition to an Embedded Trainer Controller (ETC) on board the howitzer and within the AFCS, the IFCST and IMT are used for troop operational training and maintenance training.

Detailed descriptions of the AFCS and general information regarding the HIP management, acquisition strategy, development, coordinated tests, and plans for support are contained in the Computer Resources Management Plan (CRMP) for HIP (ref 8).

DISCUSSION

All software for HIP was developed in a structured manner and documented in accordance with the government requirements to insure integrity, quality, and maintainability (ref 1). The U.S. Department of Defense standard Ada programming language was used in over 90% of the code for the AFCS. Quality requirements were applied and enforced to keep the complexity and size of individual modules to a minimum. The software was tested at various levels. Key contributing elements in this successful program were early up-front user requirement definitions, Ada, the use of software metrics like McCabes Cyclomatic Complexity, cooperation among contractors and government personnel, and a strict Independent Verification and Validation (IV&V) program implemented by the government. One key was an intensive field stress test which shook out major system errors not detected by normal testing.

IV&V is a process and a major activity of the total quality program (both software and hardware) to independently assess/confirm compliance with requirements. (Specific IV&V tasks and details for HIP are described in ref 8.)

Within the design/code verification process of the IV&V program for HIP, the software complexity for all delivered software was assessed. The automated CAT was used, as well as manual graphical techniques, to monitor contractor performance in meeting the cyclomatic complexity requirement of not exceeding nine (9).

However, due to multiple contracts for HIP to address funding and program issues, the software complexity requirement was described to meet specific areas of code.

For example, for new code, the software cyclomatic complexity in accordance with NBS 500-99 (ref 7) shall not exceed a complexity of seven (7) for the Program Design Language (PDL) and nine (9) for source code, derived from implementation of the PDL. The complexity limit of twelve (12) shall not be exceeded for software units which have been redesigned as a result of software trouble report corrective action. Software trouble reports are normally written if a software engineer discovers a problem within the software life cycle process and during software testing.

For modified code for previously developed units exceeding a complexity limit of ten (10), modifications to each of these units shall be implemented so as to achieve a complexity growth not exceeding 15%. For previously developed units with a complexity limit of ten (10) or less, modification to each of these units shall be implemented so as to achieve a complexity limit of not more than twelve (12).

Additionally, in the computation of software complexity, the Case Statement shall have complexity equal to an IF statement (e.g., complexity = 1).

MULTIPLE CONTRACTS

During initial phases of the HIP, a contract definition period clarified the software requirements for the AFCS. This period involved many meetings with the prime contractor, subcontractors, and both the U.S. and Israeli government representatives.

The HIP project progressed into Full Scale Engineering Development (FSED) where extensive software documentation reviews were held, audits were performed, and multiple meetings were conducted to ensure that the prototype software performed in accordance with the requirements. During this development phase, extensive testing was performed at the Computer Program Configuration Item (CPCI) level and Unit level as well as System Integration level. Bench level testing integrated each CPCI into the AFCS. Later in the testing process, field level testing was performed. As a result, the AFCS, PDIU, and DRU, as well as other components, were integrated into a fully functional howitzer.

The cyclomatic complexity requirement enabled testing to become easier and more manageable since a minimum amount of test paths were required in order to qualify each CPCI. In addition, the CAT produced graphical representatives of the testing paths so that any retesting or regression testing for each CPCI could be accomplished in a timely and accurate manner.

Since the FSED contract placed a lot of emphasis on government reviews, software documentation, testing, and strict controls on the cyclomatic complexity limits, future work will be minimized to make software enhancements and retest each CPCI. Thus, the future versions of production software for each CPCI will become easier to maintain, upgrade, and test as requirements from users change to meet military needs. Life cycle software maintenance, supportability, field reprogramming, and efforts required to complete IV&V's will become simplified due to the up-front efforts performed during FSED.

CONCLUSIONS

Software costs for military projects are rising each year and methods in defining software requirements are important to reduce ambiguity. In addition to clear, concise requirement definition of software products, it is also extremely important to reduce supportability and maintenance costs. Enhancements and upgrades to software can sky rocket if good controls, measures, and software tools are not available during the early phases of start-up projects.

The HIP project combined some of these ideas and concepts in a successful software design from development to limited production. The research performed on the application of software test tools (refs 7 and 9) and the development of CAT (ref 3) which was used on HIP provided a qualitative and quantitative measure of software quality. The cyclomatic complexity metric and CAT improved the structure, robustness, testability, and maintainability of the software.

The complexity metric will be a useful tool for software upgrades to meet military user's needs since regression testing, redesign work, and maintenance will become easier with the graphical aids and reduced testing paths. This metric in combination with Ada, IV&V, and government/contractor reviews should be useful in reducing software costs and maintenance.

Follow-on study should be performed to quantify these benefits in terms of time, schedule, labor, and cost. A longitudinal study which also looks at correlations between cost savings versus error rate, reliability, and modifiability would be a very useful research project for military applications and commercial use.

REFERENCES

1. MIL-STD-1679, Weapon System Software Development, December 1, 1978.

2. DARCOM-R 70-16, Management of Computer Resources in Battlefield Automated Systems, July 16, 1979.

3. Janusz, Paul E., "The Complexity Analysis Tool," Technical Report ARPAD-TR-88005, ARDEC, PAD, Picatinny Arsenal, NJ, October 1988.

4. McCabe, J.T., "A Complexity Measure," IEEE Trans. Software Engineering, Vol SE-2, No. 4, December 1976.

5. Shooman, Martin L., "Software Engineering: Design, Reliability, and Management," McGraw Hill, New York, NY, 1983.

6. Sorkowitz, Alfred, "Software Quality Assurance and Testing," presented at the Testing Computer Software Conference, Washington, DC, October 1984.

7. U.S. Department of Commerce, National Bureau of Standards, "Structured Testing: A Software Testing Methodology Using the Cyclomatic Complexity Metric," NBS Special Publication 500-99, December 1982.

8. Computer Resources Management Plan (CRMP) for the Howitzer Improvement Plan (HIP), PM HIP Joint U.S. Army-Israeli Defense Force (Fire Support Armament Center - FSAC), Picatinny Arsenal, NJ, May 1989.

9. Janusz, Paul E.; Turoczy, William R.; and Shooman, Martin L., "The Application of Software Test Tools," 1985 Proceedings Annual Reliability and Maintainability Symposium.